

Н. Б. Шаховська, Х. Ю. Гірак
Національний університет “Львівська політехніка”,
кафедра інформаційних систем та мереж

ШКАЛЮВАННЯ ЕМОЦІЙНО ЗАБАРВЛЕНИХ СЛІВ ДЛЯ ВИКОРИСТАННЯ У МЕТОДАХ КЛАСИФІКАЦІЇ ТОНАЛЬНОСТІ

© Шаховська Н. Б., Гірак Х. Ю., 2017

Запропоновано методи шкалювання емоційно забарвлених слів, що охоплюють ранжування слів, визначення коефіцієнта важливості за допомогою методики Фішберна, парне порівняння, гіпотезу Пурто тощо. Вони усі відрізняються коефіцієнтами, нормами, використанням логарифмічних шкал, хоча їхнім завданням є визначення порядку слів, фраз без глибокого аналізу їхньої тональності, емоційного забарвлення і відношення між ними. В результаті розроблено платформу для розрахункової інтегрованої оцінки, яка дасть змогу визначити думку користувача, автора тощо.

Ключові слова: емоційне забарвлення, забарвлені слова, тональні словники, валентність слова, частотний аналіз, шкала Фішберна, парне порівняння, гіпотеза Пурто.

This paper is devoted to solve a task of ranging emotive words using methods of tone classification in order to analyze author's opinion and to effectively perceive useful information from the Internet data. These methods include word-ranging, determination of importance using Fishburne method, pair comparison, hypothesis of Purto and others. They all differ in coefficients, norms, using logarithmic scales though their task is to find out the sequences of words / phrases without deep analysis of their tone, emotional color, and the relationship between them. As a result, platform has been prepared for computing integrated value, which will allow to make opinion mining of user's profile, author etc.

Key words: emotional color, emotive words, affective lexicons, valency, Fishburne method, pair comparison, hypothesis of Purto.

Оброблення великих обсягів текстових даних користувачів соціальних мереж з метою з'ясування думки автора потребує доволі багато часу і зусиль. Тим більше, якщо необхідно проаналізувати тисячі й мільйони користувачьких профілів. Ця проблематика підштовхнула до розроблення методів та засобів аналізу семантичного забарвлення текстів дописів у соціальних мережах. Такі методи дадуть змогу аналізувати думку авторів, класифікувати їх за певними критеріями, прогнозувати майбутню поведінку і потреби.

Постановка проблеми

У міру розвитку Інтернету проблема інформаційного перевантаження стає все серйознішою. Користуючись перевагами, що надає Інтернет, люди отримують інформацію в дуже великому обсязі. Актуалізується завдання ефективного і точного видобування корисної інформації з величезного обсягу даних Інтернету.

Opinion mining (sentiment analysis, дослівно “пошук думок”, “аналіз почуттів”) – широкий набір методів контент-аналізу в комп'ютерній лінгвістиці, які призначені для автоматизованого виявлення “суб'єктивної” інформації (думок, оцінкових суджень, емоцій, почуттів тощо). Метод аналізу тональності – це знаходження думок у тексті й визначення їх властивостей. Залежно від поставленого завдання нас можуть цікавити різні властивості, наприклад: а) автор – кому належить цей контент (статті, думці); б) тема – про що говориться в цьому контенті; в) тональність – позиція автора щодо теми (зазвичай “позитивна” або “негативна”) [1].

Зауважимо, що цей набір методів тісно пов'язаний з поняттям автоматизованого реферування. Автоматизоване реферування – це процес видобування найважливішої інформації з одного або декількох джерел для складання їхньої скороченої версії.

Сьогодні найдосліджуванішими є задачі класифікації тональності текстових даних. Однією з проблем є автоматичне визначення емоційного забарвлення (позитивний, негативний, нейтральний) текстових даних, тобто аналізу тональності (sentiment analysis). Важкість аналізу тональності спричинена емоційно збагаченою мовою – сленгом, багатозначністю, невизначеністю, сарказмом, і відповідно всі ці фактори вводять в оману не лише людей, але і комп'ютери.

Узагальнивши, можна розділити усі підходи до класифікації тональності на такі категорії [2]:

- Підходи, основані на правилах.
- Підходи, що ґрунтуються на словниках (тональні словники).
- Підходи, основані на шаблонах.
- Частотні методи.
- Машинне навчання (навчання без вчителя).

Підходи, основані на тональних словниках (affective lexicons) використовують список слів зі значенням тональності (валентності) кожного слова. Найпоширеніший спосіб подання документа(ів) в задачах комп'ютерної лінгвістики і пошуку – подання у вигляді:

- набору слів (bag-of-words);
- набору N-грам;
- вектором.

Наприклад, речення “я люблю чорну каву” можна подати у вигляді набору уніграм (я, люблю, чорну, каву) або біограм (я люблю, люблю чорну, чорну каву).

Інший спосіб подання тексту – символічні N-грами. Текст з прикладу можна подати у вигляді 4-символьних N-грам: “я лю”, “люб”, “юблю”, “блю” тощо. Цей спосіб подання слів тексту потребує значно довшого (повільнішого) комп'ютерного опрацювання, але має такі переваги: наявності орфографічних помилок у тексті – набір символів у тексті з помилками і набір символів у тексті без помилок практично однаковий, на відміну від слів; для мов з багатою морфологією (наприклад, для української) – в текстах можуть траплятися однакові слова, але в різних варіаціях (різний рід або число), але корінь слів не змінюється, а відповідно і загальний набір символів [8].

Сучасні методи векторного подання текстової інформації є розвитком моделей векторних просторів VSM (Vector Space Models), запропонованих у [10]. У цих моделях компоненти текстів, такі як:

- слова,
- словосполучення,
- фрагменти текстів,
- цілі документи,

подано багатовимірними векторами. Елементи векторів – це значення деякої функції від частоти виявлення компонентів текстів і їхніх контекстів. Ступінь подібності між компонентами текстів q і d визначається величиною подібності між їхніми векторами q і d .

Документ у векторній моделі розглядається як невпорядкований набір термів. Термами в інформаційному пошуку називають слова, з яких складається текст. Інколи документи, з яких виділяють терми, називають “мішком слів”, оскільки послідовність слів у документі ігнорується.

Різними засобами можна визначити вагу терма в документі – “важливість” слова для ідентифікації цього тексту. Наприклад, можна просто підрахувати кількість вживань терма в документі, так звану частоту терма, – чим частіше слово вживається в документі, тим більша у нього буде вага. Якщо терма немає в документі, то його вага в цьому документі дорівнює нулю.

Усі терми, що використано в документах певної колекції, можна впорядкувати. Після цього для деякого документа можна виписати за порядком вагу всіх термів, з тими, яких немає в цьому документі. У результаті отримуємо вектор, який і представлятиме заданий документ у векторному просторі. Розмірність цього вектора, як і розмірність простору, дорівнює кількості різних термів у всій колекції і є однаковою для всіх документів. Формально у [10] подано: $d_j = (w_{1j}, w_{2j}, \dots, w_{nj})$,

де d_j – векторне представлення j -го документа; w_{ij} – вага i -го терма в j -му документі; n – загальна кількість різних термів у всіх документах колекції.

Маючи в своєму розпорядженні такі дані для всіх документів, можна, наприклад, знаходити відстань між точками простору і тим самим вирішувати проблему подібності документів – що ближче розташовані точки, то схожіші відповідні документи. У задачах пошуку документа за запитом запит теж подається вектором того ж простору. Так можна обчислювати відповідність документів до запиту.

Для повного опису векторної моделі для пошукової системи необхідно вказати, як саме розраховуватиметься вага терма в документі. Існує декілька стандартних способів визначення функції зважування:

- tf (term frequency, частота терма) – вага визначається як функція від кількості входжень терма в документ;
- $tf-idf$ (term frequency – inverse document frequency, частота терма – обернена частота документа) – вага визначається як добуток функції від кількості входжень терма в документ та функції від величини оберненої кількості документів колекції, в яких міститься цей терм;
- $TF * IDF$, де TF – кількість входжень терма в документ, IDF – рідкість терма в колекції.
- булеве зважування – значення функції ваги дорівнює 1, якщо терм є в документі і 0 – у іншому випадку;

Зазначену модель використовують практично усі пошукові системи.

Але тут постає питання: як визначається тональність кожного слова? Також необхідно зауважити, що залежно від предметної області контексту тональність слів змінюється. Отже, завдання визначення значення тональності слова (коефіцієнта важливості, валентості) сьогодні актуальне.

Для оцінювання важливості слів широко використовують частотні методи. Вага окремого слова визначається як добуток частоти його входжень до цього документа (TF – term frequency) та ступеня важливості слова в контексті колекції (IDF – inverse document frequency):

$$TF = \frac{n_t}{\sum_k n_k}$$

де n_t – кількість входжень слова t в документ, а в знаменнику – загальна кількість слів у цьому документі [6].

Облік IDF зменшує вагу широкоживаних слів. Для кожного унікального слова в межах конкретної колекції документів існує тільки одне значення IDF .

$$IDF = \log \frac{|D|}{|\{d_i \in D | t \in d_i\}|},$$

де $|D|$ – кількість документів у корпусі; $|\{d_i \in D | t \in d_i\}|$ – кількість документів із колекції D , в яких трапляється t (коли $n_t \neq 0$) [4].

Частота входження слова оцінює важливість слова у межах цього документа, визначається як співвідношення входжень цього слова до загальної кількості слів документа.

У результаті велику вагу отримують слова з високою частотою входження у межах аналізованого документа та з низькою частотою входжень в інших документах.

Частотні методи абсолютно можуть використовуватися в автоматизації аналізу тексту. Нечастотні методи потребують залучення зовнішньої думки, як правило, експерта. Автор хоче звернути власне увагу на застосування моделі визначення лінійних вагових коефіцієнтів, а це дасть змогу виконати класифікацію через шаблон розуміння слів.

Для визначення коефіцієнта важливості слова автор пропонує підхід усереднення результатів методів:

$$P(B | X_1, X_2, \mathbf{K}, X_n) = \frac{P_1(X_n | B) \cdot P(B | X_1, X_2, \mathbf{K}, X_{n-1})}{P(X_n)}$$

Під час розгляду задач байєсівського навчання щодо параметрів багатовимірного нормального розподілу було доведено, що за необмеженого збільшення кількості показів адаптивний байєсівський фільтр наближається до оптимального фільтра.

Основні труднощі застосування байєсівського підходу пов'язані з визначенням апостеріорних ймовірностей. Як показує практика, деякі класи розподілів мають чудову властивість: у разі збільшення кількості спостережень вигляд апостеріорного закону $P(B | X_1, X_2, \mathbf{K}, X_n)$ не змінюється й збігається з апіорним розподілом $P(B)$. На кожному кроці навчання здійснюється лише перерахунок параметрів цих розподілів. Такі розподіли називаються "відтворювальними". До них, зокрема, належать: нормальний розподіл, біноміальний, Пуассона, розподіл Релея й Уїшарта.

Розв'язання цієї задачі в загальнішому випадку показує, що кінцевий обчислювальний алгоритм для рекурентного визначення $P(B | X_1, X_2, \mathbf{K}, X_n)$ може бути побудований тільки в тих випадках, коли розподіли $f(B | X_1, X_2, \mathbf{K}, X_n)$ мають достатні статистики у вигляді r -вимірного вектора результатів спостережень $P(B | X_1, \mathbf{K}, X_n)$, що можна записати

$$P(B | X_1, \mathbf{K}, X_n) = P(B, B^* | X_1, X_2, \mathbf{K}, X_n) \cdot g(X_1, \mathbf{K}, X_n),$$

де $g(X_1, \mathbf{K}, X_n)$ не залежить від B ; B^* – оцінка невідомого сигналу B на підставі вибірки.

Доведена теорема, яка стверджує, що відтворювальна апостеріорна щільність $P(B | X_1, \mathbf{K}, X_n)$ існує тоді й тільки тоді, коли спостереження $P(B | X_1, \mathbf{K}, X_n)$ допускають достатню статистику.

У вирішенні завдань, пов'язаних з класифікацією текстів, байєсівський класифікатор перевершує багато інших алгоритмів. Завдяки цьому цей алгоритм широко застосовують для фільтрації спаму (ідентифікація спаму в електронних листах) і аналізу тональності тексту (аналіз соціальних медіа, ідентифікація позитивних та негативних думок клієнтів).

6. Лінійні класифікатори. У лінійному класифікаторі категорії подано у вигляді вектора $c_i = \langle w_{1i}, \mathbf{K}, w_{ri} \rangle$ з того ж r -вимірного простору, що і вектори документів. Тоді функцію належності $CSV_i(d_j)$ можна визначити на основі скалярного добутку вектора документа і вектора категорії $\sum_{k=1}^r w_{ki} \cdot w_{kj}$. У випадку нормалізованих векторів скалярний добуток зводиться до знаходження косинуса кута між векторами [12]:

$$S(c_i, d_j) = \cos(a) = \frac{\sum_{k=1}^r w_{ki} \cdot w_{kj}}{\sqrt{\sum_{k=1}^r w_{ki}^2} \cdot \sqrt{\sum_{k=1}^r w_{kj}^2}}$$

7. Метод Роккіо. Цей метод побудований на адаптованій до теорії класифікації текстів широковідомій формулі Роккіо для релевантного зворотного зв'язку в моделі векторного простору.

Метод Роккіо визначає класифікатор $\vec{c}_i = \langle w_{1i}, \mathbf{K}, w_{ri} \rangle$ для категорії c_i :

$$w_{kj} = b \cdot \sum_{\{d_j \in POS_i\}} \frac{w_{kj}}{|POS_i|} - g \cdot \frac{w_{kj}}{|NEG_i|},$$

де w_{kj} – вага ознаки t_k у документі d_j ; $POS_i = \left\{ d_j \in T_r \mid \bigvee \Phi(d_j, c_i) = T \right\}$;

$NEG_i = \left\{ d_j \in T_r \mid \bigvee \Phi(d_j, c_i) = \bar{T} \right\}$; b і g – керуючі параметри.

8. Метод k-nearest neighbors приймає рішення про зарахування фрагмента d_j до категорії тональності c_i , k -NN проглядає k найсхожіших на d_j фрагментів, і якщо більшість з них належить до тієї ж категорії, приймається позитивне рішення. Математично класифікація документа за тональністю за допомогою k -NN зводиться до розрахунку:

$$CSV_i(d_j) = \sum RSV(d_j, \bar{d}_z) \cdot ca_{iz}$$

де $Tr_k(d_j)$ – множина k документів \bar{d}_z , на яких досягається максимум $RSV(d_j, \bar{d}_z)$, ca_{iz} – значення з коректної матриці прийняття рішення. Своєю чергою, $RSV(d_j, \bar{d}_z)$ – це деяка міра схожості між тестовим документом d_j та документом \bar{d}_z , що навчається; для цих цілей може бути використана будь-яка функція схожості або імовірнісна чи векторна міра з пошукової системи.

9. Метод опорних векторів (support vector machine – SVM) вибирає середній елемент з “найширшої” множини паралельних вирішальних поверхонь, тобто такий елемент, мінімальна відстань до якого від будь-якого навчального елемента є найбільшою. Вирішальна поверхня визначається малим набором зразків документів певної тональності, які називаються опорними векторами. У загальному випадку розглядається задача навчання з учителем: $\langle X, Y, y, X^l \rangle$, де X – простір векторів атрибутів, Y – множина категорій, $\hat{y} : X \rightarrow Y$ – цільова залежність, значення якої відомі тільки на об’єктах навчальної вибірки $X^l = (x_i, y_i)_{i=1..l}$, $y_i = \hat{y}(x_i)$. Необхідно побудувати алгоритм $a : X \rightarrow Y$, що апроксимує цільову залежність на усьому просторі X . Зазвичай розглядають задачу класифікації на два непересічні класи, в якій документи описуються n -вимірними дійсними векторами: $X = \mathbb{R}^n, Y = \{-1, +1\}$.

10. Метод моделювання максимальної ентропії (maximum entropy modeling – EM) – це інструмент об’єднання інформації з багатьох гетерогенних інформаційних джерел для класифікації. Дані на вході рішення класифікаційної проблеми описуються як множина ознак, ці ознаки можуть бути доволі складними і дозволяють досліднику використовувати апріорні знання про те, які типи інформації можуть бути корисними для класифікації. Кожна ознака відповідає обмеженню моделі. Потім обчислюється модель максимальної ентропії.

11. Word2Vec – це готовий інструмент (набір алгоритмів) для розрахунку векторних представлень слів, реалізує дві основні архітектури – Continuous Bag of Words (CBOW) і Skipgram. На вхід подається корпус тексту, а на виході отримуємо набір векторів слів.

Набір цих алгоритмів застосовується для:

- кластеризації слів за принципом їх семантичної близькості;
- кластеризація запиту;
- оцінка важливості слів у запиті;
- виявлення семантичної близькості слів;
- аналіз тональності тексту.

Послідовність роботи алгоритмів така:

1. Зчитують корпус і розраховують, наскільки часто слова виявлено в корпусі (тобто кількість разів, коли слово вжито в корпусі, для кожного слова).
2. Масив слів сортують за частотою (слова зберігають у хеш-таблиці), і видаляють рідкісні слова (їх ще називають гапаксами).
3. Будують дерево Хаффмана. Дерево Хаффмана (Huffman Binary Tree) часто застосовують для кодування словника – це значно знижує розрахункову і часову складність алгоритму.
4. Із корпусу зчитують так звані субречення (sub-sentence) і проводять субсемпсування найчастотніших слів (sub-sampling). Субречення – це деякий базовий елемент корпусу, зазвичай просто речення, але це може бути і абзац, наприклад, або навіть ціла стаття. Субсемплінг – це

процес видобування найчастотніших слів із аналізу, що пришвидшує процес навчання алгоритму і тим самим сприяє значному поліпшенню якості отриманої моделі.

5. По субреченні проходять вікном (розмір вікна задано алгоритму як параметр). У цьому випадку під вікном розуміємо максимальну дистанцію між поточним словом і словом, що передбачається в реченні. Тобто якщо вікно дорівнює три, то для речення “Я твоя хата труба хитав” аналіз здійснюватиметься всередині блока в три слова – для “Я твоя хата”, “твоя хата труба” і т.д. Вікно за замовчуванням дорівнює п’яти, рекомендованим значенням вважається десять.

6. Застосовують нейромережу прямого поширення (Feedforward Neural Network) з функцією активації ієрархічної софтмакс (Hierarchical Softmax) і/або негативне семпсування (Negative Sampling).

Загалом, CBOW і Skip-gram – це нейромережеві архітектури, що описують, як саме нейромережа “вчиться” на даних і “запам’ятовує” представлення слів. Принципи у обох архітектур різні. Принцип роботи CBOW – передбачення слова за певного контексту, а skip-gram навпаки – передбачення контенту, якщо задано слово (рис. 1).

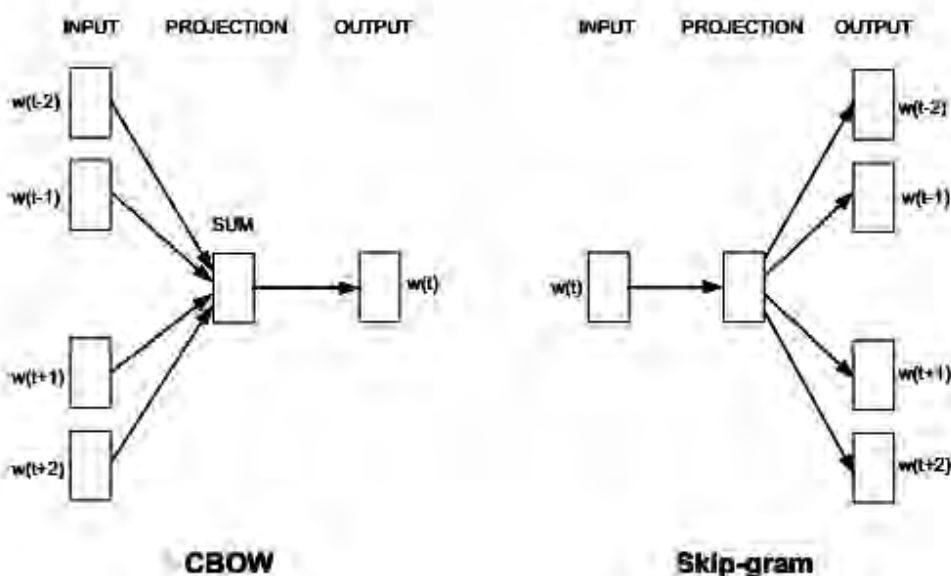


Рис. 1. CBOW і Skip-gram нейромережеві архітектури

Continuous Bag of Words (CBOW) – звичайна модель мішка слів з урахуванням чотирьох ближніх сусідів терміналу (два попередні й два наступні слова) без урахування їх послідовності.

k-skip-n-gram – це послідовність довжиною n, елементи якої розміщені на відстані не більше ніж k один від одного [9].

Як висновок:

- CBOW працює швидше, але Skip-gram працює краще, особливо для порівняно рідкісних слів;
- ієрархічний софтмакс добре підходить для створення кращої моделі порівняно рідкісних слів, негативне семпсування краще моделює частотніші слова;
- застосування субсемпсування підвищує продуктивність. Рекомендований параметр субсемпсування від $1e-3$ до $1e-5$;
- чим більший розмір вектора, тим краще (але це не весь час залежить від корпусу);
- розмір вікна – для Skip-gram оптимальний розмір близько 10, для CBOW – близько 5.

Отже, в інформаційному пошуку найпоширенішим методом оцінювання ваги ознаки (слова) є TF-IDF. Для аналізу тональності цей метод не дає хороших результатів. Причина цього – для аналізу тональності не настільки важливі слова, які часто повторюються в тексті (тобто слова з високим TF), на відміну від задачі пошуку. Тому зазвичай використовують бінарну вагу, тобто ознакам (словам) присвоюють одиничну вагу, якщо вони є в тексті. В іншому випадку вага дорівнює нулю.

Опис розробленого алгоритму

1. Попереднє оброблення тексту.

На цій стадії видаляються всі html теги (якщо працюємо з гіпертекстом), знаки пунктуації, символи. Ця операція реалізується на переписаній зовнішній бібліотеці мови програмування php “Beautiful Soup”. Надалі в тексті є так звані “стоп-слова” – це часті слова в мові, які переважно не мають ніякого сенсового навантаження (наприклад, в англійській мові це такі слова, як “the, at, about ...”). Стоп-слова видаляють за допомогою пакета NLTK. Після опрацювання вихідного тексту отримують набір слів (масив слів). На цьому етапі можна і далі покращувати структуру слів, але це буде виконано у подальших дослідженнях. Тобто припустимо, що подано всього три дописи соцмережі (скажімо, коментування новин) з такими попередньо обробленими векторами слів:

§ [performance, roof, automobile]

§ [autopilot, capacity, battery]

§ [seats, decor, speed]

2. Подання у виді вектора.

Для цього розглянуто методи ранжування слів і відповідно складено базис-словник, враховуючи експертні оцінки для подальших досліджень. Отже, на цьому кроці потрібно подати текст у вигляді вектора із чисел (вибірково можна використати словники Даля або Залізняка), тобто замінити слова із тексту індексом з попередньо розробленого словника. Для більшої наочності пропонується об'єднати всі слова із списку в пункті 1 і відсортувати самостійно методом шкали Фішберна:

[automobile, battery, speed, autopilot, performance, capacity, decor, seats, roof]

Замінюючи попередні вектори на індекс слова в словнику, отримуємо:

§ [1,0,0,0,1,0,0,0,1]

§ [0,1,0,1,0,1,0,0,0]

§ [0,0,1,0,0,0,1,1,0]

Одержані вектори називаються “вектори слів” або ж “feature vector”. Отже, отримуємо вектори для кожного припису соцмережі.

3. Знаходження інтегрального значення.

Маючи відшкаловані вектори слів, і, відповідно, розраховані вагові коефіцієнти важливості кожного слова, можна переходити до визначення інтегральної оцінки кожного припису, а також до класифікації текстів. Такі дослідження заплановано провести у подальших наукових роботах.

Було розроблено програмний модуль “Semant-1”, який реалізовує описаний вище алгоритм; цей модуль здатний ефективно використовуватися для задач семантичного ранжування слів (рис. 2).

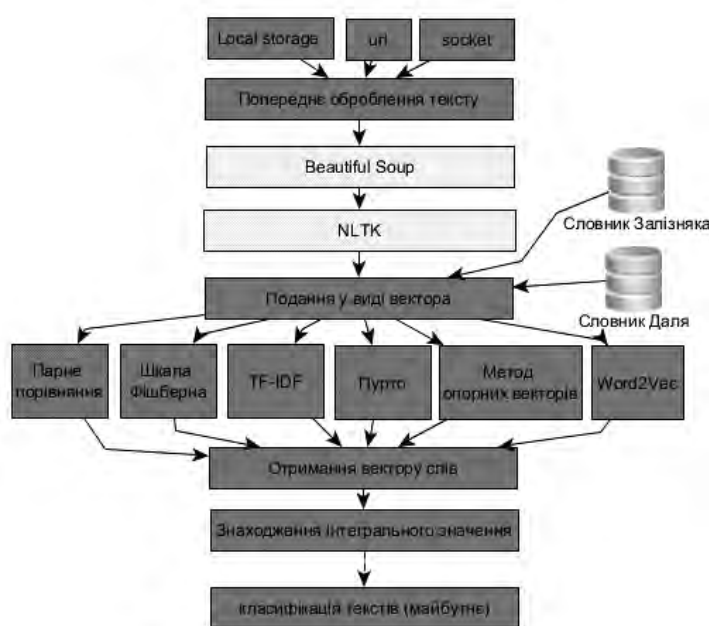


Рис. 2. Структурна схема програмного модуля “Semant-1”

Висновок

Існує велика кількість розроблених і успішно впроваджених методів для частотного аналізу текстів і входження у них окремих слів. Вони відрізняються коефіцієнтами, нормуваннями, використанням логарифмічних шкал. Але їх завдання – “сире” знаходження послідовностей слів/фраз, без глибшого аналізу їх тональності, емоційного забарвлення, взаємовідношень між ними, тому в статті проаналізовано сучасні нечастотні способи визначення валентності слів і можливості щодо їх застосування в інформаційних системах. В результаті підготовлено підґрунтя для розрахунку інтегральної оцінки окремих сутностей (речень), що в майбутньому дасть змогу зробити сумарний висновок щодо профілю користувача, автора публікації тощо.

1. Pang B. *Opinion Mining and Sentiment Analysis* / B. Pang, L. Lee // *Foundations and Trends in Information Retrieval: Vol. 2. No. 1–2, 2008.* 2. Данилюк І. Г. *Технологія автоматичного визначення тематики тексту [Текст] / І. Г. Данилюк // Лінгвістичні студії: зб. наук. пр. Вип. 17 / уклад.: Анатолій Загнітко (наук. ред.) та ін. – Донецьк : ДонНУ, 2008. – С. 290–293.* 3. Литвин В. В. *Метод квазіреферування текстових документів на основі онтології предметної області / В. В. Литвин, Т. І. Черна, В. М. Ковалевич // Відбір і обробка інформації, Вип. № 41(117). – 2014. – С. 100–108.* 4. Медиковський М. О. *Дослідження ефективності визначення вагових коефіцієнтів важливості / М. О. Медиковський, О. Б. Шуневич // Вісник Хмельницького національного університету. – 2011. № 5. – С. 176–182.* 5. Хомів Б. А. *Компаративний аналіз математичних моделей, методів та засобів оцінювання opinii в текстових даних інтернет-ресурсів / Б. А. Хомів, С. А. Лупенко, А. С. Сверстюк // Вісник Хмельницького національного університету. – 2011. – № 6. – С. 7–16.* 6. Чалая Л. Э. *Меры важности концептов в семантической сети онтологической базы знаний [Текст] / Л. Э. Чалая, Ю. Ю. Шевякова, А. Ю. Шафроненко // Матеріали другої міжнар. наук.-техн. конф. “Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління”. – К. : КДАВТ, 2011. – С. 51.* 7. Шаховська Н. Б., Нога Р. Ю. *Аналітичний огляд методів та засобів опрацювання текстової інформації // Інформаційні системи та мережі. – № 715. – Л. : Вид-во Львівської політехніки, 2011. – С. 215–223.* 11. Wu H. and Luk R. and Wong K. *Interpreting TF-IDF term weights as making relevance decisions // ACM Transactions on Information Systems, 26 (3). 2008.* 12. Katrin ERK. *Vector space models of word meaning and phrase meaning: A survey. Language and Linguistics Compass, 2012, 6.10: 635–653.* 8. *Інтернет-ресурс TF-IDF. Режим доступу: [https://ru.wikipedia.org/wiki/TF-IDF].* 9. *Інтернет-ресурс Okapi. – Режим доступу: [https://ru.wikipedia.org/wiki/Okapi_BM25].* 10. *Інтернет-ресурс. – Режим доступу: [https://habrahabr.ru/post/149605/].* 11. *Інтернет-ресурс. – Режим доступу: [http://nlpx.net/archives/179].*